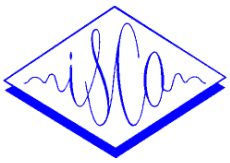

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

NOVEL TWO-PASS SEARCH STRATEGY USING TIME-ASYNCHRONOUS SHORTEST-FIRST SECOND-PASS BEAM SEARCH

Atsunori Ogawa, Yoshiaki Noda, and Shoichi Matsunaga

NTT Cyber Space Labs.
1-1 Hikari-no-oka, Yokosuka-shi, Kanagawa 239-0847 Japan
{ogawa, noda, mat}@nttspch.hil.ntt.co.jp

ABSTRACT

In this paper, we describe a novel two-pass search strategy for large vocabulary continuous speech recognition. The first-pass of this strategy uses a regular time-synchronous beam search with rough models to generate a word lattice. Then, the second-pass search derives exact results from the word lattice using more accurate models. This search is "time-asynchronous shortest-first beam search", which has two novel features: a time-asynchronous beam search mechanism using heuristics that are scores on the word lattice nodes and a strict pruning scheme using shortest-first hypothesis extension. 20k-word Japanese broadcast news recognition experiments show that our second-pass search is more accurate and more efficient than either N-best rescoring or A* search that are conventional second-pass search methods.

1. INTRODUCTION

Large vocabulary continuous speech recognition is very expensive computationally. One of the most common search strategies for large vocabulary continuous speech recognition is the multi-pass search strategy, which employs more accurate and expensive models in later search stages [1]. Multi-pass search strategies can reduce the amount of computation without decreasing accuracy. The two-pass search is one of the most popular multi-pass search strategies. In the two-pass search, the second-pass search works on the intermediate form generated by the first-pass search. There are various types of two-pass search strategies [2,3,4]. We also have explored the two-pass search strategy on NTT continuous speech recognition system VoiceRex [5,6].

In this paper, we focus on a two-pass search strategy that employs a regular time-synchronous beam search in the first-pass and a word lattice as the intermediate form. We developed a novel second-pass search that works on the word lattice. The search is time-asynchronous shortest-first beam search which has two novel features: a time-asynchronous beam search mechanism using heuristics that are scores on the word lattice nodes and a strict pruning scheme using shortest-first hypotheses extension.

In the following sections, we describe the details of our second-pass search method and give the results of a continuous speech recognition experiment to show the efficiency of our method in comparison with other conventional second-pass search methods.

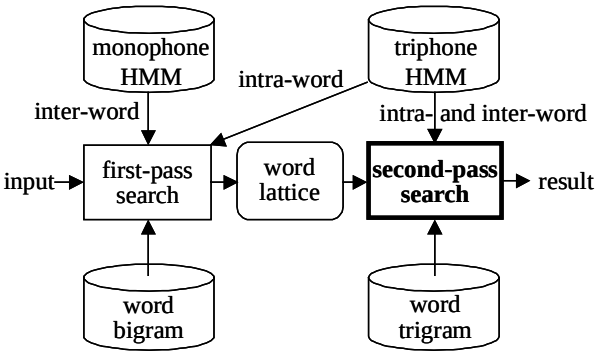

Figure 1: Two-pass search strategy in NTT LVCSR.

2. SYSTEM OVERVIEW

The outline of the two-pass search strategy employed in NTT recognition system is shown in Figure 1. In the first-pass, a regular time-synchronous beam search using rough models such as word bigrams, intra-word triphone HMMs, and inter-word monophone HMMs is carried out to generate a word lattice. In the second-pass, a more accurate search using models such as word trigrams, and intra- and inter-word triphone HMMs is carried out to derive exact results from the word lattice.

There are various second-pass search methods that work on a word lattice. Commonly used ones are, for example, N-best rescoring [7] and A* search [4]. We will describe these conventional methods first to show how our time-asynchronous shortest-first beam search differs from them.

3. CONVENTIONAL SECOND-PASS SEARCH

3. 1. N-best Rescoring

N-best rescoring extracts the N most-likely hypotheses from the word lattice and recalculates their scores using more accurate language models. The N-best rescoring strategy is easy to implement, but acoustic rescoring using more accurate acoustic models is difficult to carry out because it explosively increases the processing time. Without acoustic rescoring, the recognition results are not always accurate enough. Furthermore, because the N most-likely hypotheses are often word sequences differing by only one word, we must set N to be large enough to ensure accurate recognition results.

3. 2. A* Search

Figure 2 shows the A* search procedure in the second-pass. In the A* search, the highest-score hypothesis on the stack (list of hypotheses sorted by score) is extended in a right-to-left direction (best-first search). The score of the hypothesis n at time t is defined as follows:

$$\hat{f}_n(t) = g_n(t) + \hat{h}_n(t) \qquad (1)$$

where $g_n(t)$ is the second-pass search score calculated using more accurate language and acoustic models (word trigrams, and intra- and inter-word triphone HMMs), and $\hat{h}_n(t)$ is the estimated score of the unsearched space (heuristics). The first-pass scores on the word lattice nodes can be used as heuristics. If the heuristics are equal to the real scores in the second-pass, an efficient and accurate search is realized, and optimal recognition results are obtained.

However, the models used in the second-pass are usually more accurate than those of the first-pass, so the first-pass scores are not always best for use as heuristics. This causes an explosive increase in the number of hypotheses (stack size). Consequently, in the A* search, sometimes it is difficult to obtain recognition results without having to limit the size of the stack.

4. TIME-ASYNCHRONOUS SHORTEST-FIRST SECOND-PASS BEAM SEARCH

To solve the problems of the conventional second-pass search methods described in section 3, we developed a time-asynchronous shortest-first beam search.

4. 1. Basic Algorithm

Figure 3 shows our second-pass search procedure. Hypotheses are evaluated with equation (1) and extended in the right-to-left direction using first-pass scores on the word lattice nodes as heuristics, like in the A* search. However, the hypotheses extension is done in the breadth-first manner, and the hypotheses are pruned by limiting the stack size, like in a regular time-synchronous beam search. In our search, recognition results are not always optimal because of the hypotheses pruning. But, they are certainly obtained for any input utterances. Furthermore, to increase the efficiency of the search, the heuristics can be weighted to make them closer to the real score of the second-pass search. This algorithm is based on the beam search method using heuristics which is described in [8].

4. 2. Score Based Pruning

In addition to hypotheses number based pruning, score threshold based pruning is also carried out. Hypotheses number based pruning is carried out only after each hypothesis extension finishes. On the other hand, score threshold based pruning is carried out at every frame (every Viterbi calculation step on the HMM state). In other words, these two pruning techniques are complementary, and together, they increase the efficiency of the hypotheses pruning.

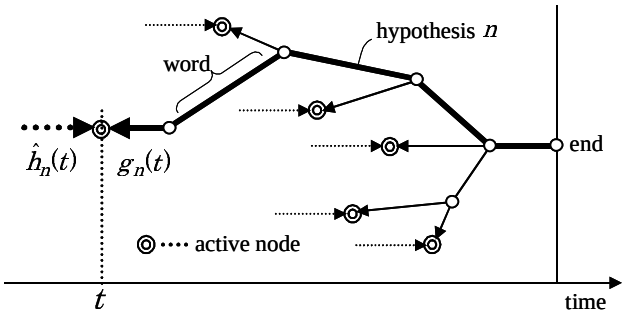

Figure 2: A* search procedure.

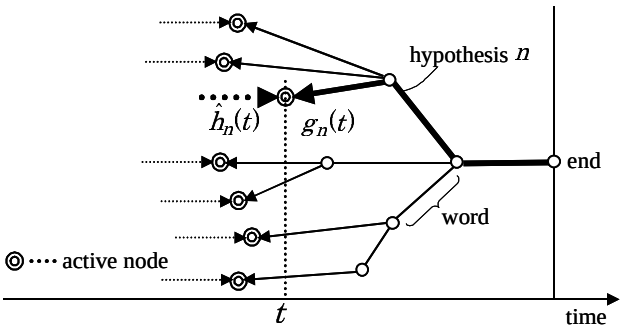

Figure 3: Proposed search procedure.

The score based pruning mechanism is shown in Figure 4. During the second-pass search, the maximum second-pass scores of each frame are stored to make the maximum second-pass score envelope. The pruning threshold score envelope is generated whose values stay fixed below the maximum second-pass score of each frame. Then, for each frame, the hypotheses whose scores are lower than the threshold set by the pruning score envelope are pruned. To do this pruning efficiently, the shortest hypothesis on the stack is extended first (shortest-first search). By extending the hypotheses in the shortest-first manner, the length of each hypothesis becomes nearly equal (see Figure 3), and the pruning threshold score envelope can be accurately estimated, like in a regular time-synchronous beam search.

4. 3. Word Boundary Time Lag

Because the models used in the second-pass search are usually more accurate than those used in the first-pass search, the word boundary times stored on the word lattice nodes (in Figure 5, t_1 and t_2) are not always best for the second-pass search. If the second-pass hypotheses assign word boundaries according to the word lattice, those scores would be estimated to be lower than the actual values.

To improve hypothesis score accuracy in the second-pass, we consider the word boundary time lag between the first-pass and the second-pass search and set free boundary intervals of a few frame widths on both sides of the boundary times of the word lattice nodes. The

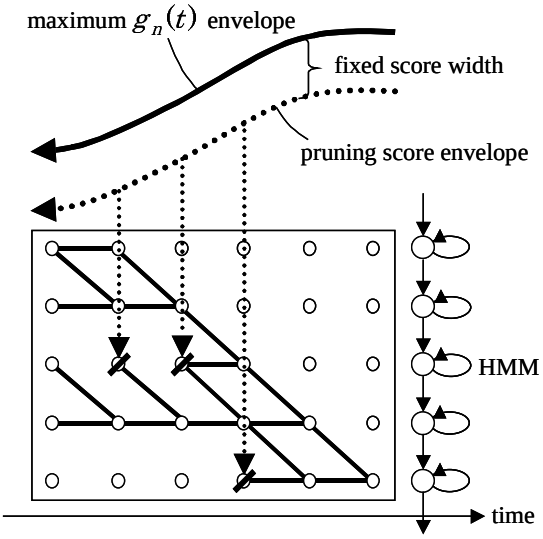

Figure 4: Score threshold based pruning.

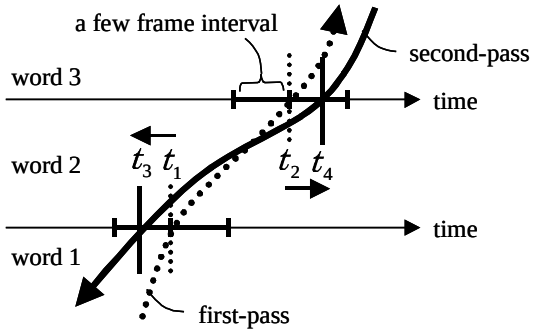

Figure 5: Word boundary time lag.

second-pass hypotheses can assign word boundaries anywhere within these intervals. In Figure 5, the word boundary time between word 1 and 2 moves from t_1 to t_3, and the one between word 2 and 3 moves from t_2 to t_4.

5. EXPERIMENTAL RESULTS

To evaluate our method, we carried out 20k-word Japanese broadcast news speech recognition experiments.

5. 1. Conditions

We prepared monophone and triphone HMMs, trained by using about 6,700 utterances that had been collected from TV broadcast news programs. The monophone HMMs had 120 states, and each state had 16 continuous density Gaussian mixture components. The triphone HMMs had 2,000 shared-states, and each state had 8 continuous density Gaussian mixture components. Sampling frequency was 16 kHz, frame length was 30 msec, and frame period was 10 msec. The feature vector consisted of 12 Mel-scaled FFT-based cepstrum (MFCC) coefficients, logarithmic power, and their first and second derivatives. The total number of parameters for each vector was 39.

We also prepared word bigram and trigram models, trained by using about 500,000 TV broadcast news manuscripts.

The evaluation set consisted of 50 clean utterances by 5 male speakers from TV broadcast news (1,800 words and 12 sec average length). Its perplexity was 105.8 when using word bigram, and 57.0 when using word trigram. The out-of-vocabulary rate was 0.6%.

The basic recognition system was VoiceRex, which was developed at NTT Cyber Space Labs [9]. The experiments were carried out on a Sun Ultra Enterprise 450 (UltraSPARC-II 296 MHz) workstation.

The word accuracy of the first-pass search was 90.49%.

Table 1: Effect of pruning.

num. hypo.	score	proc. time [sec]	word acc. [%]
1000	OFF	19.10	93.29
1000	ON	5.00	93.34
250	OFF	4.02	93.06
250	ON	1.78	93.34

Table 2: Effect of word boundary time lag.

width [msec]	proc. time [sec]	word acc. [%]
0	1.78	93.34
10	1.84	93.90
20	1.77	93.96
30	1.79	93.85
40	1.95	93.85
50	1.89	93.57

5. 2. Effect of Pruning

First, we examined the effect of hypotheses number based and score threshold based pruning. Here, we did not consider the word boundary time lag described in section 4.3. We used four pairs of pruning thresholds and measured their average processing times for the second-pass search and their word accuracies.

The results are shown in Table 1. Table 1 shows that the two pruning methods contribute to the efficiency of the second-pass search.

5. 3. Effect of Word Boundary Time Lag

Next, we examined the effect of the word boundary time lag. From the results of the previous section, we decided to keep the number of hypotheses under 250 and to use score threshold based pruning. We used the word boundary intervals of from 0 msec to 50 msec in width (from 0 frames to 5 frames).

The results are shown in Table 2. Table 2 shows that, as a result of including the word boundary time lag, the recognition accuracies have slightly improved with no increase in processing time. In this case, the recognition accuracy was most improved when the interval width was 20 msec.

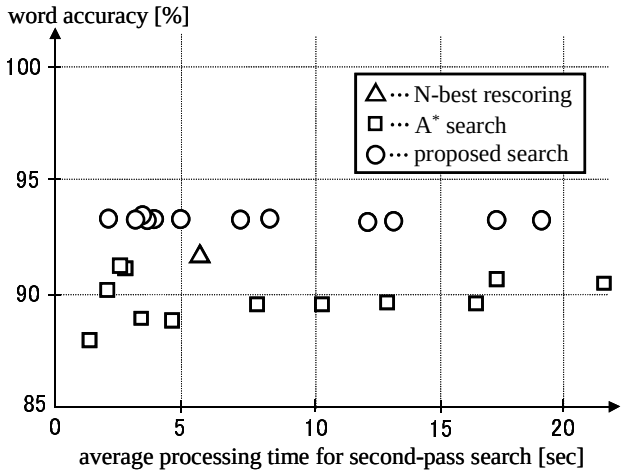

Figure 6: Comparison of the three methods.

Table 3: Best performances of the three methods.

search	proc. time [sec]	word acc. [%]
N-best	5.37	92.45
A*	2.22	92.11
proposed	1.77	93.96

5. 4. Comparison of Three Methods

Finally, we compared our method with the N-best rescoring and A* search methods. For the N-best rescoring experiment, N was set to 300. Because the pure A* search described in section 3.1 caused a stack size explosion, we had to employ some kind of pruning technique. We chose the hypotheses number based and score threshold based pruning, that is the same pruning techniques as in our method. The word boundary time lag was also taken into account. We set various pruning thresholds including the thresholds given in section 5.2.

The results are shown in Figure 6 and in Table 3. In Figure 6, the horizontal axis represents the average processing time for the second-pass search, and the vertical axis represents the word accuracy. The best performances of the three second-pass search methods are shown in Table 3. Figure 6 and Table 3 collectively show that our method is more accurate and more efficient than the other two second-pass search methods. In the A* search, the accuracy and efficiency of the search greatly depends on the quality of the input utterance (or the heuristics). On the other hand, the accuracy and efficiency of our method does not depend on the quality of the input utterance, and recognition results were obtained certainly for any input utterances.

6. SUMMARY

We have described the time-asynchronous shortest-first second-pass beam search for the two-pass search strategy. This search has two novel features: a time-asynchronous beam search mechanism using heuristics that are scores on the word lattice nodes and a strict pruning scheme by shortest-first hypothesis extension. Moreover, to take the word boundary time lag between the first-pass and the second-pass search into account, the search uses a free word boundary interval of a few frame widths. The results of the 20k-word Japanese broadcast news recognition experiments show that our search method is more accurate and more efficient than either the N-best rescoring or A* search methods.

7. ACKNOWLEDGEMENTS

The authors would like to thank NHK (Japan Broadcast Corporation) for providing us with the broadcast news database. The authors are also grateful to the members of the Speech Recognition Group of the Media Processing Group for their comments and cooperation to this research. The authors would like to thank Dr. Kazuhiko Yamamori, the executive manager of our project, and Dr. Takeshi Kawabata, the leader of our group, whose support made our research possible.

8. REFERENCES

1. R. Schwartz, L. Nguyen, and J. Makhoul, "Multiple-Pass Search Strategies," *Automatic Speech and Speaker Recognition Advanced Topics*, Kluwer Academic Publishers, pp.429-456, 1996.
2. S. Ortmanns, and H. Ney, "A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition," *Computer Speech and Language*, Vol.11, No.1, pp.43-72, 1997.
3. S. Austin, R. Schwartz, and P. Placeway, "The Forward-Backward Search Algorithm," *Proc. ICASSP*, Vol.1, pp.697-700, 1991.
4. E. -F. Huang, F. K. Soong, and H. -C. Wang, "The Use of Tree-Trellis Search for Large-Vocabulary Mandarin Polysyllabic Word Speech Recognition," *Computer Speech and Language*, Vol.8, No.1, pp.39-50, 1994.
5. Y. Noda, S. Matsunaga, and S. Sagayama, "An Approximation Technique in Large-Vocabulary Continuous Speech Recognition Using a Word Graph," *Technical Report of IEICE*, SP96-102, pp.53-58, 1997 (in Japanese).
6. A. Ogawa, Y. Noda, and S. Matsunaga, "A Second-Pass Search Algorithm for Multi-Pass Speech Recogntion Strategy," *Technical Report of IPSJ*, SLP30-11, pp.51-56, 2000 (in Japanese).
7. F. Richardson, M. Ostendorf, and J. R. Rohlicek, "Lattice-Based Search Strategies for Large Vocabulary Speech Recognition," *Proc. ICASSP*, Vol.1, pp.411-414, 1995.
8. Y. Noda, S. Sagayama, "Fast and Accurate Beam Search Using Forward Heuristic Functions in HMM-LR speech Recognition," *Proc. Eurospeech*, Vol.2, pp.913-916.
9. Y. Noda, Y. Yamaguchi, K. Ohtsuki, A. Ogawa, S. Nakagawa, A. Imamura, "The Development of Speech Recognition Engine VoiceRex," *Proc The 1999 Autumn Meeting of The Acoustical Society of Japan*, Vol.1, pp.91-92, 1999 (in Japanese).